

Routing-Free Mixture-of-Experts

Yilun Liu^{*,1,3}, Jinru Han^{*,2}, Sikuan Yan¹, Volker Tresp^{1,3}, Yunpu Ma^{1,3}

¹ Ludwig Maximilian University of Munich ² University of California, Los Angeles

³ Munich Center for Machine Learning * *Equal contribution.*

yilun.liu@tum.de jinruhan1219@ucla.edu cognitive.yunpu@gmail.com

Abstract

Standard Mixture-of-Experts (MoE) designs rely heavily on centralized routing mechanisms that introduce rigid inductive biases. We propose Routing-Free MoE, which eliminates all hard-coded centralized components, including external routers, Softmax, TopK, and fixed load balancing. All related functionalities become encapsulated within individual experts, allowing each expert to determine its activation entirely on its own. We introduce a unified, adaptively scheduled load-balancing framework to simultaneously optimize both expert-balancing and token-balancing objectives through a customizable interpolation, letting optimal resource allocation pattern emerge locally during training. Extensive experiments demonstrate that Routing-Free MoE achieves consistently better performance, efficiency, and robustness than baseline approaches at both training and deployment. We further analyze its emergent behavior in detail and offer insights that may facilitate future MoE design and optimization. Code is available at <https://liuyilun2000.github.io/RoutingFreeMoE/>.

1 Introduction

The scalability of transformer-based Large Language Models (LLMs) (Vaswani et al., 2017) is constrained heavily by their substantial computational demands (Kaplan et al., 2020). To efficiently expand model capacity without proportional growth in computation, Mixture-of-Experts (MoE) designs focus on activating subsets of parameters for each input (Shazeer et al., 2017; Fedus et al., 2022). This approach presents a fundamental challenge: how to optimally distribute inputs to experts while satisfying efficiency considerations.

Existing MoE designs are hindered by structural limitations across multiple dimensions. Standard MoE relies on small, external routers that lack sufficient capacity for storing expert capabilities to determine expert preference for each input, forcing

them to learn their prediction via indirect trial-and-error optimization, which inevitably leads to suboptimal routing and unstable early training dynamics (Lv et al., 2025). For computational efficiency, standard MoE enforces rigid, global constraints on expert activation that ignore input-specific dynamics. The fixed TopK selection imposes uniform sparsity, regardless of varying input complexity and expertise of experts, overlooking potential gains from fewer or more activations (Zhou et al., 2022). The Softmax operation forces a competitive probability distribution that sacrifices the absolute magnitude information of expert activations (Wang et al., 2024b). For load-balancing, common approaches employ different balancing targets that force predetermined activation patterns, which are often mutually exclusive yet exhibit varying performance under different configurations (Fedus et al., 2022; Zhou et al., 2022; Muennighoff et al., 2025). Strict adherence to either may constrain the model’s ability to adaptively discover potentially better resource allocation patterns (Do et al., 2026).

To bridge these gaps, we introduce Routing-Free MoE, a bottom-up architecture without any centralized routers, Softmax, TopK, or rigid load-balancing designs, where each expert individually and directly determines its own activation. For efficiency, we design a unified, configurable framework to adaptively achieve the sparsity and load-balancing objectives together during training, allowing optimal activation pattern to emerge spontaneously at each individual expert. We utilize an auxiliary loss function that interpolates both token- and expert-balancing, providing flexibility to adaptively exploit both depending on training dynamics and workload demands.

We validate Routing-Free MoE across three scales with extensive pretraining experiments in comparison against baselines. Routing-Free MoE consistently achieves better language modeling quality and downstream performance averaged

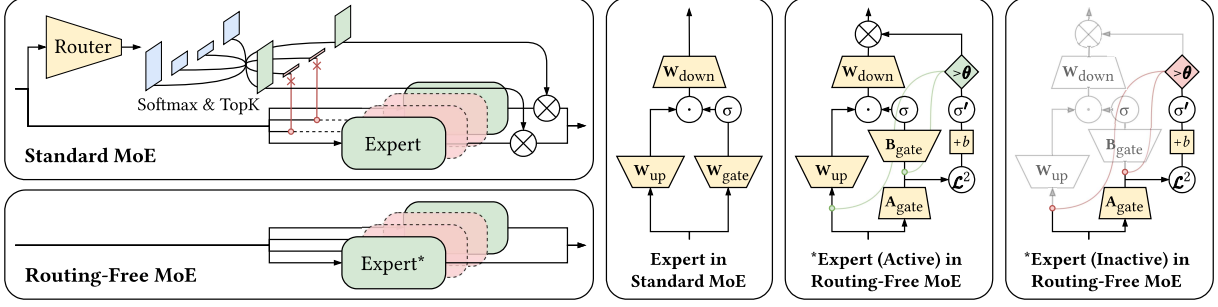


Figure 1: Standard MoE relies on routing to orchestrate expert activations. We let each expert independently determine its own activation. Green indicates activated components; red for inactive ones; yellow for trainable ones.

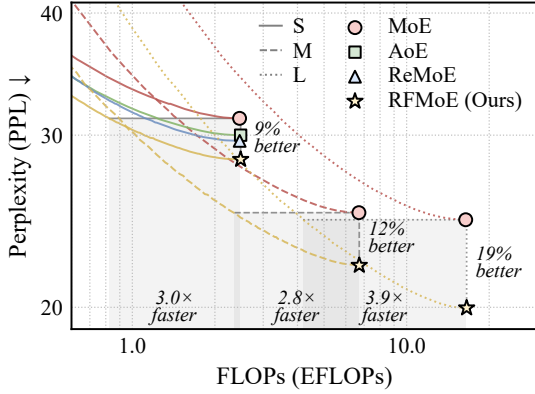


Figure 2: Routing-Free MoE consistently outperforms standard MoE, AoE (Lv et al., 2025), and ReMoE (Wang et al., 2024b) in language modeling. All models are trained on OpenWebText (Gokaslan et al., 2019) under identical environment conditions and best-performing configurations, as described in Section 4.1. FLOPs are estimated for one epoch.

across 9 evaluation benchmarks, with improved scalability, robustness, efficiency, and flexibility at both training and deployment. We further analyze its training dynamics in detail, and document emergent phenomena of density and load-balancing, offering insights that may help guiding future MoE design and deployment.

In summary, our key contributions are:

- A Routing-Free MoE architecture that eliminates routers, Softmax, TopK, and hard-coded load balancing mechanisms.
- A unified, adaptive load-balancing framework that jointly optimizes token-balancing and expert-balancing, allowing customizable application.
- Experiments and analyses demonstrating the improvements of Routing-Free MoE over baselines.

2 Preliminaries

Consider a standard MoE LLM (Fedus et al., 2022; Jiang et al., 2024) comprising L transformer blocks. For layer $\ell \in \{1, \dots, L\}$ and token sequence of

length T , the forward pass¹ can be formulated as:

$$\mathbf{x}_{1:T}^\ell = \text{SelfAttn}(\mathbf{h}_{1:T}^{\ell-1}) + \mathbf{h}_{1:T}^{\ell-1}, \quad (1)$$

$$\mathbf{h}_t^\ell = \text{MoE}(\mathbf{x}_t^\ell) + \mathbf{x}_t^\ell, \quad (2)$$

where $\mathbf{x}_{1:T}^\ell$ denotes the attention module output with residual connection added. The MoE block performs a token-wise mapping, yielding output $\mathbf{h}_t^\ell$ at token $t \in \{1, \dots, T\}$ with residual added.

The mainstream form of MoE LLMs incorporates multiple structurally identical FFN experts $E_i(\cdot)$, $i \in \{1, \dots, N\}$ within the MoE layer. This is implemented via a token-wise routing mechanism among all N experts at layer ℓ , with a gating network $G(\cdot)$ assigning each token to a designated number K of top-activated experts:

$$\mathbf{h} = \sum_{i=1}^N (G(\mathbf{x})_i E_i(\mathbf{x})) + \mathbf{x}, \quad (3)$$

$$G(\mathbf{x}) = \text{Softmax}(\text{TopK}(\mathbf{x}\mathbf{G}, K)), \quad (4)$$

where $G(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}^N$ denotes the routing mechanism, whose output serves as the weights of the weighted sum for outputs among all N experts, and only K of N experts receive nonzero values. The router learns its weight matrix $\mathbf{G} \in \mathbb{R}^{D \times N}$ that can be interpreted as a set of N individual D -dimensional expert vectors $\mathbf{g}_i$, each responding to a characteristic hidden state $\mathbf{h}_i$ that should activate the corresponding expert E_i , with $\mathbf{s}_i = \mathbf{x}\mathbf{g}_i \in \mathbb{R}^N$ denoting the token-to-expert affinity scores (Zhou et al., 2022; Liu et al., 2026b). $\text{TopK}(\mathbf{s}, K)$ retains the top- K scores and masks the rest to $-\infty$.

Each individual expert E_i is implemented as a Feed-Forward Network (FFN). Modern FFN typically takes the form of a Gated Linear Unit (GLU, Dauphin et al. (2017); Shazeer et al. (2017)):

$$\text{FFN}(\mathbf{x}) = [\sigma(\mathbf{x}\mathbf{W}_{\text{up}}) \odot (\mathbf{x}\mathbf{W}_{\text{gate}})]\mathbf{W}_{\text{down}}, \quad (5)$$

where σ is the activation function and $\odot$ denotes element-wise multiplication. In MoE LLMs, this

¹LayerNorms and dropout are omitted for clarity.

design enables a second level of input-dependent information filtering, leading to potentially more effective representations.

The routing in standard MoE relies on several strong inductive biases. With merely N expert vectors $\mathbf{g}_i \in \mathbb{R}^D$, the router’s capacity is orders of magnitude smaller than the experts themselves, yet must compress the knowledge-intensive activation preferences of all N experts into single dot-product scores without any direct signal about expert capabilities, improving only through indirect trial-and-error (Lv et al., 2025). TopK hard-codes a fixed sparsity ratio K/N regardless of input complexity, preventing input-adaptive activation patterns (Zhou et al., 2022). Softmax discards absolute activation magnitudes by forcing a competitive probability distribution, suppressing the residual contribution of highly suitable experts when others also happen to score higher (Wang et al., 2024b).

3 Methodology

3.1 Architecture

To alleviate the router bottleneck, Lv et al. (2025) introduce Autonomy-of-Experts (AoE), with

$$\text{FFN}(\mathbf{x}) = [\sigma(\mathbf{x}\mathbf{A}_{\text{gate}}\mathbf{B}_{\text{gate}}) \odot (\mathbf{x}\mathbf{W}_{\text{up}})]\mathbf{W}_{\text{down}}, \quad (6)$$

where $\mathbf{A}_{\text{gate}} \in \mathbb{R}^{D \times r}$ projects the input hidden state $\mathbf{x}$ from hidden dimension D to a lower-dimensional rank $r \ll D$, and $\mathbf{B}_{\text{gate}} \in \mathbb{R}^{r \times D_{\text{act}}}$ projects it back. This low-rank representation provides an alternative indicator of expert suitability that originates from within the expert itself rather than from an external router. Each expert can therefore directly produce its own scalar activation score by applying a norm to its own internal representation $\|\mathbf{x}\mathbf{A}_{\text{gate},i}\|_2$. However, AoE feeds the internal scores $\|\mathbf{x}\mathbf{A}_{\text{gate}}\|_2$ back to the standard centralized TopK and Softmax routing pipeline:

$$G(\mathbf{x}) = \text{Softmax}(\text{TopK}(\|\mathbf{x}\mathbf{A}_{\text{gate}}\|_2, K)), \quad (7)$$

thereby retaining the structural constraints and inductive biases of conventional routing.

Meanwhile, addressing the constraints of TopK and Softmax, Wang et al. (2024b) propose ReMoE, which replaces TopK and Softmax with a single ReLU function applied directly to router’s output:

$$G(\mathbf{x}) = \text{ReLU}(\mathbf{x}\mathbf{G}), \quad (8)$$

The sparse activation arises naturally from ReLU without any explicit TopK selection or comparative normalization. The absolute magnitude of router scores is preserved, allowing each expert’s residual contribution to be linearly weighted by router’s

prediction, rather than a normalized relative preference. Nevertheless, ReMoE still retains a centralized external router, preserving the information bottleneck and indirect optimization dynamics.

Building on these insights, our Routing-Free MoE seeks to eliminate all constraints of routing mechanisms. We adopt AoE’s FFN design (Equation 6) using each expert’s internal norm $\|\mathbf{x}\mathbf{A}_{\text{gate}}\|_2$ as the initial expert preference score, grounding the activation decision in the expert’s own response to the input. Since $\|\mathbf{x}\mathbf{A}_{\text{gate}}\|_2$ is strictly non-negative unlike router’s $\mathbf{x}\mathbf{G}$, we introduce a learnable per-expert bias term before activation upon ReMoE’s design, yielding

$$G_i(\mathbf{x}) = \text{ReLU}(\|\mathbf{x}\mathbf{A}_{\text{gate},i}\|_2 - b_i). \quad (9)$$

With the per-expert bias term introduced, experts whose weighted norm falls below their own bias threshold contribute zero and are effectively deactivated, allowing each expert to jointly adapt both its $\mathbf{A}_{\text{gate},i}$ matrix and the b_i parameter to effectively adjust its own activation ratio. We further introduce a global post-activation threshold θ as a configurable hyperparameter for external control over the overall sparsity level, giving the final binary activation decision of each expert:

$$f_i(\mathbf{x}) = \mathbb{1} \{G_i(\mathbf{x}) - \theta \geq 0\}. \quad (10)$$

The result is a fully decentralized architecture without any external router, TopK or Softmax, making each expert independently determine its own activation from within, allowing the global activation pattern to emerge bottom-up from collective self-adjustment of all experts. Figure 1 visualizes the architecture of Routing-Free MoE and its experts.

3.2 Training

Training MoE models requires simultaneously maintaining the activation ratio and balanced expert and token distribution. Standard practice hard-codes TopK for the activation ratio, and addresses balancing via either *Token-Choice* (TC) (Shazeer et al., 2017; Fedus et al., 2022) that guarantees per-token compute but not expert balance, or *Expert-Choice* (EC) (Zhou et al., 2022) that hard-codes uniform expert utilization but not per-token compute, as illustrated in Figure 3. Both enforce one balanced dimension as a hard constraint and optimizes the other via soft auxiliary loss. As Routing-Free MoE has eliminated all centralized routing mechanisms, standard activation ratio and load-balancing strategies that rely on hard-coded TopK

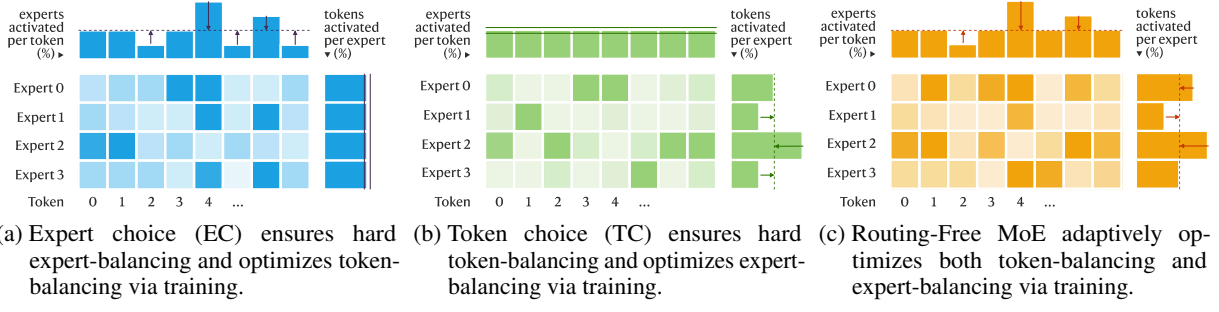


Figure 3: Load-balancing for tokens and experts. Routing-Free MoE introduces a unified load-balancing framework that simultaneously optimizes both expert-balancing and token-balancing through a configurable interpolation.

(Do et al., 2026) no longer apply. We introduce a unified load-balancing framework by extending the auxiliary loss of Fedus et al. (2022) to jointly encourage both balanced token distribution across experts and balanced expert activation per token, without requiring any centralized mechanisms.

As the binary activation decision f_i in Equation 10 is non-differentiable, we directly use the pre-threshold activation weight $G_i(\mathbf{x})$ as a differentiable activation proxy of expert E_i . We define the mean activation density over a set of experts $\mathcal{E}$ and token batch $\mathcal{B}$ and its differentiable proxy as:

$$\rho(\mathcal{E}, \mathcal{B}) = \frac{1}{|\mathcal{E}||\mathcal{B}|} \sum_{e_i \in \mathcal{E}} \sum_{\mathbf{x} \in \mathcal{B}} f_i(\mathbf{x}), \quad (11)$$

$$\tilde{\rho}(\mathcal{E}, \mathcal{B}) = \frac{1}{|\mathcal{E}||\mathcal{B}|} \sum_{e_i \in \mathcal{E}} \sum_{\mathbf{x} \in \mathcal{B}} G_i(\mathbf{x}). \quad (12)$$

Both quantities equal the target activation density ρ_∞ under perfectly uniform load.

We decompose the load-balancing objective into two complementary terms². The *expert-balancing* loss $\mathcal{L}_{\text{EB}}$ encourages uniform distribution of tokens across experts by penalizing experts that consistently receive more or fewer tokens than average:

$$\mathcal{L}_{\text{EB}} = \frac{1}{|\mathcal{E}|} \sum_{e_i \in \mathcal{E}} \left(\frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} f_i(\mathbf{x}) \right) \left(\frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} G_i(\mathbf{x}) \right). \quad (13)$$

The *token-balancing* loss $\mathcal{L}_{\text{TB}}$ encourages uniform distribution of activated experts per token by penalizing tokens that consistently activate more or fewer experts than average:

$$\mathcal{L}_{\text{TB}} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} \left(\frac{1}{|\mathcal{E}|} \sum_{e_i \in \mathcal{E}} f_i(\mathbf{x}) \right) \left(\frac{1}{|\mathcal{E}|} \sum_{e_j \in \mathcal{E}} G_j(\mathbf{x}) \right). \quad (14)$$

Each loss is a dot product between a binary non-

differentiable term and a differentiable proxy, and is minimized when both factors equal ρ_∞ uniformly³. The two objectives are combined via a configurable interpolation parameter $\mu \in [0, 1]$:

$$\mathcal{L}_{\text{LB}} = \mu \mathcal{L}_{\text{EB}} + (1 - \mu) \mathcal{L}_{\text{TB}}. \quad (15)$$

Setting $\mu = 1$ recovers a pure expert-balancing, and $\mu = 0$ recovers pure token-balancing. This provides a single unified framework that interpolates both routing paradigms, allowing load-balancing to be tailored to specific deployment needs.

Rather than fixing the auxiliary loss coefficient λ as a static hyperparameter, we follow Wang et al. (2024b) and adopt an adaptive schedule that drives the empirical activation ratio toward the target ρ_∞ throughout training. The total training objective is

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda_t \mathcal{L}_{\text{LB}}, \quad (16)$$

with λ_t updated at each training step t as:

$$\lambda_{t+1} = \lambda_t \cdot (1 + \eta)^{\text{sign}(\rho_t(\mathcal{E}, \mathcal{B}) - \rho_\infty)}. \quad (17)$$

When the current density ρ_t exceeds the target ρ_∞ , λ_t increases to exert stronger load-balancing pressure; when below, it decreases to allow more expert activations. The step size η controls the responsiveness of this feedback loop. This formulation avoids the need to manually tune λ while ensuring the model converges to the desired computational budget, and is compatible with both training from scratch and adaptation of pretrained MoE models into Routing-Free MoE.

To encourage all experts to participate during the early warm-up phase of training, each expert’s bias is initialized as $1e^{-6}$, allowing all experts to become activated, jointly exploring the representation space and establishing initial specialization before sparsity is enforced. As training progresses and λ increases, the sparsity regularization gradually strengthens, driving the activation density toward target ρ_∞ . By this stage, experts have already de-

²We avoid using the terms expert choice and token choice in Routing-Free MoE because the notion of “choice” implies a centralized comparison and selection process, which contradicts our principle that all decisions should emerge locally and independently at individual experts and tokens.

³See Appendix A.

	Arch.	Size	FLOPs	PPL↓	PIQA	HellaS	WinoG	ARCe	ARCc	OBQA	QQP	QNLI	SST-2	Avg.
S	MoE	92.44M	90.93M	31.22	57.56	27.19	51.22	33.42	21.33	24.60	36.82	49.46	49.08	38.96
	AoE	93.85M	88.57M	30.00	56.09	27.01	50.20	33.84	21.93	25.00	36.82	49.46	49.08	38.82
	ReMoE	92.44M	90.93M	29.60	56.58	26.69	51.62	33.38	22.27	26.00	36.83	49.48	49.08	39.10
	RFMoE	95.32M	91.08M	27.42	58.49	27.09	50.59	35.77	21.59	24.40	36.98	49.44	53.56	39.77
M	MoE	289.9M	248.0M	25.00	58.32	28.26	52.33	36.20	21.42	24.60	36.85	49.46	49.31	39.64
	RFMoE	307.3M	249.2M	22.08	58.92	27.85	49.72	35.27	21.50	26.40	37.06	49.51	57.34	40.40
L	MoE	808.4M	608.4M	24.58	59.19	29.37	50.83	36.41	23.29	25.00	37.57	49.28	49.08	40.00
	RFMoE	870.6M	613.2M	19.97	58.87	28.27	50.51	37.46	21.67	26.60	39.93	49.86	53.67	40.76

Table 1: Evaluation results for standard MoE baseline and Routing-Free MoE across scales and benchmarks. Each model is trained on OpenWebText (Gokaslan et al., 2019) from scratch for one epoch. Details in Appendix E.

veloped distinct roles, avoiding expert collapse and ensuring a stable and effective phase transition.

4 Experiments

We conduct comprehensive experiments to validate the performance of Routing-Free MoE.

4.1 Experimental Setup

We implement Routing-Free MoE upon the HuggingFace implementation⁴ for the Mixtral architecture (Jiang et al., 2024), which includes attention designed as Grouped Query Attention (GQA, Ainslie et al. (2023)) with Rotary Position Embeddings (RoPE, Su et al. (2024)), FFN using SwiGLU activation (Shazeer, 2020), and Root Mean Square Layer Normalization (RMSNorm, Zhang and Sennrich (2019)) applied to the residual stream prior to each attention and MoE FFN (Xiong et al., 2020).

We conduct experiments across three scales (S/M/L) up to 0.8B, and compare against a standard MoE baseline with identical structure except for the routing mechanism. AoE (Lv et al., 2025) and ReMoE (Wang et al., 2024b) are also examined as both additional baselines and ablated variants for comparison. Detailed hyperparameters and training configurations are in Table 8 in Appendix.

Following standard practices, all models are trained on OpenWebText (Gokaslan et al., 2019) for one epoch. Zero-shot evaluation is conducted across 9 benchmarks (Gao et al., 2024b). We include PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-Easy and ARC-Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), and QQP, QNLI, and SST-2 from GLUE (Wang et al., 2018). Accuracy is reported on WinoGrande, QQP, QNLI, and SST-2, and normalized accuracy on others.

⁴<https://github.com/huggingface/transformers/tree/v4.57.6/src/transformers/models/mixtral>

Config.	Size	FLOPs	PPL↓
Standard MoE	92.44M	90.93M	31.22
w/o router (AoE, $r=16$)	93.85M	88.57M	30.00
w/o router (AoE, $r=32$)	95.32M	91.08M	30.31
w/o TopK&Softmax (ReMoE)	92.44M	90.93M	29.60
Routing-Free MoE ($r=16$)	93.85M	88.57M	28.73
Routing-Free MoE ($r=32$)	95.32M	91.08M	28.33

Table 2: Ablation performed at scale S. Detailed configurations for each run are provided in Table E.

4.2 Main Results

Table 1 presents language modeling perplexity and downstream benchmark evaluation results across all three scales under an iso-compute comparison, with FLOPs matched within $\sim 1\%$ between MoE and Routing-Free MoE at each scale. Figure 2 shows the detailed validation perplexity evolution with total FLOPs throughout training. Routing-Free MoE achieves consistently better validation perplexity than standard MoE across all scales. On downstream benchmarks, Routing-Free MoE also consistently improves the average performance, with a detailed statistical analysis (Appendix C) supporting that Routing-Free MoE can produce a statistically significant improvement over the standard MoE baseline. Notably, the gain of Routing-Free MoE does not diminish with scale, suggesting that the benefits of our approach can continue to provide gains as models scale up.

4.3 Architectural Comparison

We experiment at scale S, decomposing the contribution of each architectural change over the standard MoE via incremental ablation, with main results shown in Table 2. Routing-Free MoE outperforms either by a substantial margin under matching size and FLOPs. A visualization of perplexity curves during training for AoE and ReMoE can also be found in Figure 2.

Table 3 reports the effect of r . Increasing r consistently improves perplexity, yet with diminishing

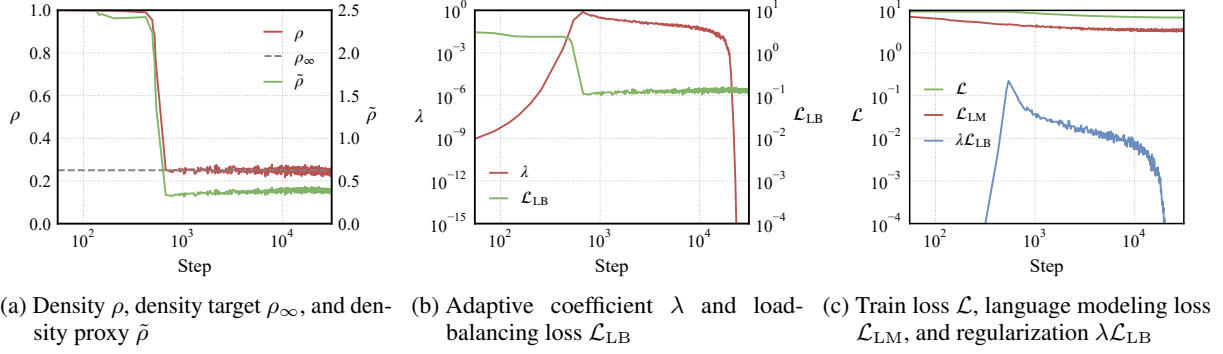


Figure 4: Training dynamics of Routing-Free MoE at scale S, with $r = 16$, $\lambda_0 = 1e^{-10}$, $\eta = 0.02$, and $\alpha = 1e^{-3}$.

r	Size	FLOPs	PPL↓
8	93.11M	87.32M	29.16
16	93.85M	88.57M	28.74
32	95.32M	91.08M	28.34
64	98.27M	96.09M	28.24

Table 3: Effect of low-rank projection dimension at scale S, with learning rate $\alpha = 1e^{-3}$.

returns. Therefore, we set 32 as the default r and scale it proportionally with the hidden dimension D across experiments on other scales for the optimal efficiency–quality trade-off.

A detailed analysis about Routing-Free MoE at deployment is provided in Appendix B, including efficiency improvements by expert parallelism and the effects of adapting threshold θ at inference.

4.4 Training Dynamics

Figure 4 visualizes the training dynamics of Routing-Free MoE. The empirical activation density ρ begins near 1 at initialization, drops sharply to the target ρ_∞ as λ grows, and remains close to the target thereafter. The differentiable proxy $\tilde{\rho}$ also settles slightly above ρ_∞ at convergence. λ rises steeply during the initial warm-up phase, plateaus as ρ converges to ρ_∞ , and collapses sharply near the end as density falls marginally below the target. The regularization term $\lambda\mathcal{L}_{LB}$ exhibits a transient spike caused by the exponential growth of λ , with its magnitude rising above 10^{-1} , making it a non-negligible contributor to the total loss $\mathcal{L}$ alongside $\mathcal{L}_{LM}$, and steering gradient descent toward directions that also reduce $\mathcal{L}_{LB}$. As training progresses, $\lambda\mathcal{L}_{LB}$ decays to a negligible level, demonstrating how load-balancing pressure is applied precisely when needed without persistently distorting the language-modeling optimization objectives.

Figure 5 further examines training stability by sweeping the learning rate α . Under conservative α , both models converge smoothly with Routing-

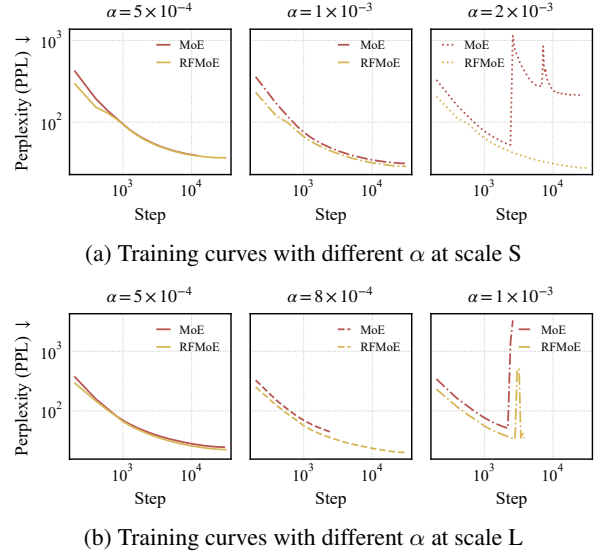


Figure 5: Training dynamics of Routing-Free MoE with different α , with $r = 16$, $\lambda_0 = 1e^{-10}$, and $\eta = 0.02$.

Free MoE consistently achieving lower perplexity. As α increases, the standard MoE baseline begins to collapse much earlier than Routing-Free MoE. This is evident at scale S, where MoE fails at $\alpha = 2e^{-3}$ while Routing-Free MoE remains stable throughout training. In addition, tuning α for the standard MoE at scale L yields only marginal gains over its scale-M performance (as in Figure 2); whereas Routing-Free MoE continues to improve as scale increases, highlighting its enhanced scalability. Routing-Free MoE not only outperforms the baseline, but also exhibits better training stability and less sensitivity to learning rates, tolerating a wider range of hyperparameter choices at scale.

5 Discussion

Beyond its performance, here we document some intriguing aspects of Routing-Free MoE.

5.1 Per-Layer and Global Density

With Top-K at each layer eliminated, a natural question is whether the target activation density ρ_∞

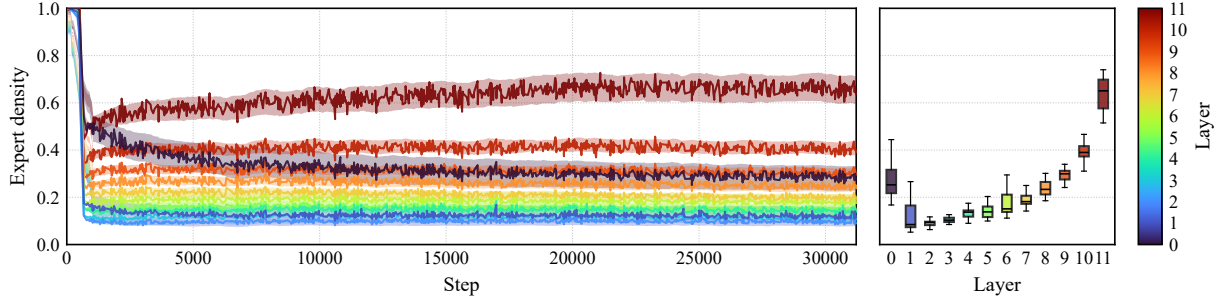


Figure 6: Evolution of expert activation across layers during training on OpenWebText under global density target. Left panel shows the per-layer mean activation as lines, along with IQR as shaded band regions using 1,000-step moving average smoothing. Right panel shows activation distribution at the final training step. Model at scale S.

	μ	PPL $\downarrow$	Eval. Throughput $\uparrow$
only TB	0.0	28.41	645.7
	0.2	28.35	648.3
balanced	0.5	28.34	662.3
	0.8	28.38	643.9
only EB	1.0	28.43	648.8

Table 4: Effect of load balancing interpolation parameter μ . Throughput is measured as samples per second.

should still be enforced within each layer, or only globally across all experts. Per-layer enforcement computes $\mathcal{L}_{LB}$ separately at each layer, continuing to impose the inductive bias for a uniform sparsity at every depth. A global ρ_∞ , in contrast, relaxes the constraint and allows individual layers to deviate from ρ_∞ so long as aggregate activation matches the target and improves the overall loss $\mathcal{L}$. Our experiment on scale S finds that relaxing this inductive bias leads to a significant improvement with perplexity drops from 39.44 to 28.74. In Figure 6 we illustrate the emergent expert activation pattern that Routing-Free MoE develops by itself during training when this per-layer bias is removed. A detailed analysis on this matter is provided in Appendix D.2. Enforcing identical sparsity at every depth suppresses the compute-hungry layers that naturally benefit from activating more experts, while simultaneously forcing unnecessary activations in layers where sparse representations suffice. Once this bias is lifted, the model is free to self-organize into a more effective, functionally aligned activation structure.

5.2 Token and Expert Balancing

A key contribution of this work is demonstrating empirically that token- and expert-balancing strategies can be combined in a complementary manner via soft interpolation rather than being treated as mutually exclusive. Prior work (Zhou et al., 2022; Muennighoff et al., 2025) documented that EC and

TC perform better under different configurations, but our unified framework provides a mechanism to adaptively balance both objectives.

Table 4 shows that $\mu = 0.5$ achieves the lowest perplexity and highest throughput, which degrades toward either direction. Token-balancing enables flexible compute allocation based on input characteristics, while expert-balancing ensures uniform expert utilization and encourages specialization; the two objectives address orthogonal failure modes and jointly produce better outcomes than either alone. This complementarity is illustrated in Figure 7. Under token-only balancing (Figure 7a), the absence of expert balancing leads to larger activation probability differences across the expert axis, with a few experts activated far more frequently than others. Conversely, under expert-only balancing (Figure 7c), expert loads are more horizontally uniform, but activation patterns vary notably across benchmarks, indicating that without token-level regularization, expert activation becomes overly sensitive to the input domain distribution. When both objectives are active ($\mu = 0.5$, Figure 7b), the activation pattern is more uniform along both axes and more consistent across inputs.

6 Related Work

6.1 MoE Foundations

Mixture-of-Experts was introduced as adaptive mixtures of local experts (Jacobs et al., 1991; Jordan and Jacobs, 1994) and later scaled to deep networks (Eigen et al., 2014; Shazeer et al., 2017). Interpretability studies reveal that FFNs in transformers capture knowledge with sparse activations (Geva et al., 2021; Dai et al., 2022; Dalvi et al., 2019; Durrani et al., 2020; Gurnee et al., 2023), motivating MoE designs that activate only a subset of parameters (Liu et al., 2023). Modern MoE architectures have since been deployed at scale

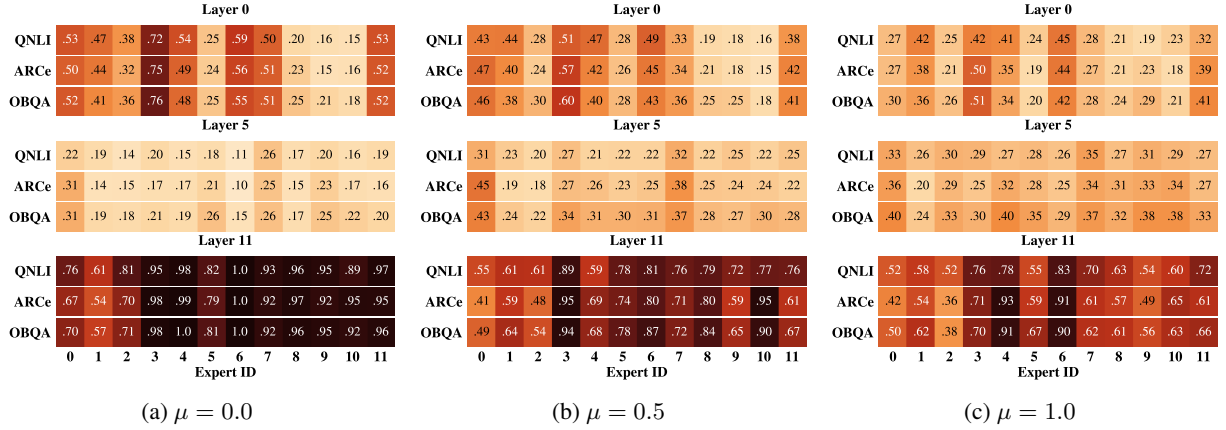


Figure 7: Expert activation heatmaps on different input for representative layers at scale S. Each row shows the average expert activation ratios on a given benchmark. Darker colors indicate more frequent activation.

(Lepikhin et al., 2021; Fedus et al., 2022; Zoph, 2022; Komatsuzaki et al., 2023; Rajbhandari et al., 2022; Du et al., 2022), with frontier models with over billions of parameters (Jiang et al., 2024; Dai et al., 2024; Grok, 2024). Structural enhancements such as shared experts (Gou et al., 2026; Dai et al., 2024) further help reduce parameter redundancy.

6.2 Routing Mechanisms

Traditional MoE relies on centralized routers, learned linear projections followed by TopK selection (Lepikhin et al., 2021; Fedus et al., 2022). Recent work has moved toward relaxing the routing mechanisms. Lv et al. (2025) replaces the router with expert-internal scoring; Wang et al. (2024b) replaces Softmax and TopK with ReLU gating. Other approaches include using vector quantization for expert assignment (Do et al., 2025), virtual shared experts (Wu et al., 2026), using pretrained language models as routers (Liu and Lo, 2025), and with ternary expert expansion (Yan et al., 2025). Huang et al. (2024) propose adjusting expert counts based on input difficulty. Do et al. (2026) utilize global TopK selection over a combined similarity score to unify token- and expert-choice routing.

6.3 Load Balancing and Training

Token Choice (Shazeer et al., 2017; Fedus et al., 2022) guarantees per-token compute but not expert balance. Expert Choice (Zhou et al., 2022) guarantees expert load balance, but may potentially yield suboptimal matching. Studies has shown that Token Choice and Expert Choice balancing are complementary rather than mutually exclusive (Muennighoff et al., 2025). Recent advances include auxiliary-loss-free balancing via dynamic bias (Wang et al., 2024a; DeepSeek-AI et al.,

2025), similarity-preserving routers (Omi et al., 2025), expert specialization through orthogonality losses (Guo et al., 2025b; Feng et al., 2025), and infrastructure-level scheduling (Zhao et al., 2026; Yu et al., 2025). For training stability, TopK selection’s discontinuity prevents gradient flow to non-selected experts; Wang et al. (2024b) address this with differentiable ReLU gating and adaptive regularization. Qiu et al. (2025) provide practical guidance on auxiliary losses, while He et al. (2024) and Pan et al. (2024) address auxiliary losses at inference. Do et al. (2026) propose a unified scoring function with global TopK selection that linearly combines TC and EC similarity scores, which addresses representation collapse and token dropping simultaneously. Elango et al. (2026) and Cui et al. (2026) explores MoE efficiency from the direction of communication overhead and parallelism. Most of the aforementioned approaches retain explicit routing mechanisms and remain dependent on centralized routing decisions.

7 Conclusion

We present Routing-Free Mixture-of-Experts, an MoE architecture that entirely eliminates centralized routing mechanisms, along with a unified adaptive load-balancing framework that jointly optimizes token- and expert-balancing objectives during training. Experiments across 3 scales and 9 benchmarks validate that Routing-Free MoE consistently outperforms baselines, with improved efficiency and robustness. We further analyze the load-balancing behavior throughout training and the expert activation patterns. Routing-Free MoE opens a path toward more flexible and efficient architectures. We hope this work encourages future exploration of MoE design and optimization.

Limitations

Our experiments are conducted exclusively on models trained from scratch on OpenWebText at scales up to 0.8B parameters. While results are consistent across three scales, it remains an open question whether the findings generalize to even larger-scale pretraining regimes, where prior scaling-law studies suggest that both performance and qualitative model behavior might shift (Kaplan et al., 2020; Hoffmann et al., 2022; Wei et al., 2022), which lie beyond our current resource limits. Similarly, we only evaluate on 9 English benchmarks for commonsense reasoning and natural language inference, which are still limited in scope relative to broader concerns raised about benchmark generality in NLP (Raji et al., 2021) and may not capture the full range of applications present in real-world applications. We leave performance in other domains such as arithmetic and logic reasoning (Cobbe et al., 2021), coding (Chen et al., 2021; Liu et al., 2026a), safety (Jiao et al., 2026), multilingual settings (Hu et al., 2020; Yang et al., 2026), low-resource languages (Hedderich et al., 2021), and other multimodal or domain-specific settings (Lin et al., 2026; Jiang et al., 2026; Yan et al., 2026) unexplored, as well as robustness under distribution shift or adversarial input (Nie et al., 2020). Given that our method follows standard training practices common among a broad body of peer-reviewed research in this field, it is highly plausible that the observed trends will carry over to broader settings.

Another practical direction we have not explored is converting an existing pretrained standard MoE model into a Routing-Free MoE via continued pretraining or training-free adaptation, rather than training from scratch, following the spirit of prior dense-to-MoE conversion methods (Zhang et al., 2022; Komatsuzaki et al., 2023; Liu et al., 2026b). A conversion could further reduce the compute cost of adopting our approach, making it a promising avenue for future work; yet it introduces a vast set of distinct challenges beyond the scope of this work.

Ethics Consideration

Routing-Free MoE is proposed as a general purpose language model architecture and does not introduce domain-specific risks beyond those inherent to language model pretraining more broadly (Bender et al., 2021; Weidinger et al., 2021; Bommasani et al., 2022). However, as with any method that improves model efficiency and scalability, it may

lower the barrier to training more capable models, with corresponding societal and governance implications that merit careful consideration in downstream applications (Strubell et al., 2019; Sastry et al., 2024).

The OpenWebText dataset used in our experiments (Gokaslan et al., 2019) may also reflect demographic or geographic biases present in large-scale web corpora, similar to those documented in other web-scraped pretraining datasets (Dodge et al., 2021; Gehman et al., 2020). We did not conduct dedicated bias or safety audits, and the absence of such analyses means that potential fairness, privacy, or safety issues may persist (Mitchell et al., 2019; Gebru et al., 2021). It is therefore incumbent upon downstream users and deployers to perform appropriate task-specific fairness, privacy, and safety evaluations before any real-world deployment. We disclaim responsibility for unintended consequences arising from downstream use that involve models trained using our approach.

Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the Karlsruhe Institute of Technology National High Performance Computing Center (NHR@KIT) under the NHR projects 22189, 22560, and 24767. The HoreKa supercomputer at NHR@KIT is funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research of Germany. We also acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LEONARDO under project EHPC-AI-2024A06-060, hosted by CINECA (Italy) and the LEONARDO consortium through a EuroHPC Regular Access call. This work receives support from the Munich Center for Machine Learning (MCML) and the Program of China Scholarship Council (Grant No.202508080292). The funding bodies had no role in the design of methodology and experiments, the analysis and interpretation of results, or the writing of manuscript. In addition, the authors thank Xu He and Alinur Kozhanov from Technical University of Munich for their contributions during the early stages of this work.

References

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. [Gqa: Training generalized multi-query](#)

- transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the dangers of stochastic parrots: Can language models be too big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. [Piqa: Reasoning about physical commonsense in natural language](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. [On the opportunities and risks of foundation models](#). *Preprint*, arXiv:2108.07258.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? try arc, the ai2 reasoning challenge](#). *ArXiv*, abs/1803.05457.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Chenwei Cui, Rockwell Jackson, Benjamin Joseph Herrera, Ana María Tárano, and Hannah Kerner. 2026. [Multi-head latentmoe and head parallel: Communication-efficient and deterministic moe parallelism](#). *Preprint*, arXiv:2602.04870.
- Damai Dai, Chengqi Deng, Chenggang Zhao, R.x. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.k. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. [DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, Bangkok, Thailand. Association for Computational Linguistics.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. [Knowledge neurons in pretrained transformers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.
- Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. [What is one grain of sand in the desert? analyzing individual neurons in deep nlp models](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317.
- Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. [Language modeling with gated convolutional networks](#). In *International conference on machine learning*, pages 933–941. PMLR.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. [Deepseek-v3 technical report](#).
- Giang Do, Hung Le, and Truyen Tran. 2026. [Rethinking sparse mixture of experts from a unified perspective](#). *Preprint*, arXiv:2503.22996.
- Giang Do, Kha Pham, Hung Le, and Truyen Tran. 2025. [On the role of discrete representation in sparse mixture of experts](#). *Transactions on Machine Learning Research*.
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, and 8 others. 2022. [GLaM: Efficient scaling of language models with mixture-of-experts](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.
- Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. [Analyzing individual neurons in pre-trained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4865–4880.

- David Eigen, Marc’Aurelio Ranzato, and Ilya Sutskever. 2014. [Learning factored representations in a deep mixture of experts](#). *Preprint*, arXiv:1312.4314.
- Venmugil Elango, Nidhi Bhatia, Roger Waleffe, Rasoul Shafipour, Tomer Asida, Abhinav Khattar, Nave Assaf, Maximilian Golub, Joey Guman, Tiyasa Mitra, Ritchie Zhao, Ritika Borkar, Ran Zilberstein, Mostofa Patwary, Mohammad Shoeybi, and Bitar Rouhani. 2026. [Latentmoe: Toward optimal accuracy per flop and parameter in mixture of experts](#). *Preprint*, arXiv:2601.18089.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#). *Journal of Machine Learning Research*, 23(120):1–39.
- Yuchen Feng, Bowen Shen, Naibin Gu, Jiaxuan Zhao, Peng Fu, Zheng Lin, and Weiping Wang. 2025. [Dive into moe: Diversity-enhanced reconstruction of large language models from dense into mixture-of-experts](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19375–19394.
- Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024a. [Higher layers need more lora experts](#). *Preprint*, arXiv:2402.08562.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024b. [The language model evaluation harness](#).
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. [Datasheets for datasets](#). *Commun. ACM*, 64(12):86–92.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [RealToxicityPrompts: Evaluating neural toxic degeneration in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.
- Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. <http://Skylion007.github.io/OpenWebTextCorpus>.
- Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T. Kwok, and Yu Zhang. 2026. [Mixture of cluster-conditional lora experts for vision-language instruction tuning](#). *IEEE Transactions on Image Processing*, 35:3881–3892.
- Grok. 2024. [Open release of grok-1](#).
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, ZF Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025a. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Xinye Cao, Sicong Leng, Qimei Cui, and Xudong Jiang. 2025b. [Advancing expert specialization for better moe](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. [Finding neurons in a haystack: Case studies with sparse probing](#). *Transactions on Machine Learning Research*.
- Xin He, Shunkang Zhang, Yuxin Wang, Haiyan Yin, Zihao Zeng, Shaohuai Shi, Zhenheng Tang, Xiaowen Chu, Ivor Tsang, and Ong Yew Soon. 2024. [Expert-flow: Optimized expert activation and token allocation for efficient mixture-of-experts inference](#). *arXiv preprint arXiv:2410.17954*.
- Michael A. Hedderich, Lukas Lange, Heike Adel, Jan-nik Strötgen, and Dietrich Klakow. 2021. [A survey on recent approaches for natural language processing in low-resource scenarios](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.
- Roger W. Hockney. 1994. [The communication challenge for mpp: Intel paragon and meiko cs-2](#). *Parallel Computing*, 20(3):389–398.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22*, Red Hook, NY, USA. Curran Associates Inc.
- Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. [XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

- Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. [Harder task needs more experts: Dynamic routing in moe models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12883–12895.
- Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. [Adaptive mixtures of local experts](#). *Neural computation*, 3(1):79–87.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. [Mixtral of experts](#). *Preprint*, arXiv:2401.04088.
- Kailin Jiang, Ning Jiang, Yuntao Du, Yuchen Ren, Yuchen Li, Yifan Gao, Jinhe Bi, Yunpu Ma, Bin Li, Lei Liu, and Qing Li. 2026. [MINED: Probing and updating with multimodal time-sensitive knowledge for large multimodal models](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 13766–13795, San Diego, California, United States. Association for Computational Linguistics.
- Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, J  rgen Pfeffer, and Ashton Anderson. 2024. [Spin: Sparsifying and integrating internal neurons in large language models for text classification](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4666–4682.
- Difan Jiao, Yilun Liu, Ye Yuan, Zhenwei Tang, Linfeng Du, Haolun Wu, and Ashton Anderson. 2026. [LLM safety from within: Detecting harmful content with internal representations](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 39711–39727, San Diego, California, United States. Association for Computational Linguistics.
- Michael I Jordan and Robert A Jacobs. 1994. [Hierarchical mixtures of experts and the em algorithm](#). *Neural computation*, 6(2):181–214.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *Preprint*, arXiv:2001.08361.
- Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. 2023. [Sparse upcycling: Training mixture-of-experts from dense checkpoints](#). In *The Eleventh International Conference on Learning Representations*.
- Dmitry Lepikhin, HyukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. [GShard: Scaling giant models with conditional computation and automatic sharding](#). In *International Conference on Learning Representations*.
- Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. 2026. [Moe-llava: Mixture of experts for large vision-language models](#). *IEEE Transactions on Multimedia*, 28:4408–4419.
- Changzhi Liu, Yilun Liu, Sikuan Yan, Volker Tresp, and Yunpu Ma. 2026a. [Mendel g  del machine: Recursive self-improving coding agents via comparative evolution](#). *Preprint*, arXiv:2608.07645.
- Kuan-Ming Liu and Ming-Chih Lo. 2025. [Llm-based routing in mixture of experts: A novel framework for trading](#). *Preprint*, arXiv:2501.09636.
- Yilun Liu, Yunpu Ma, Yuetian Lu, Shuo Chen, Zifeng Ding, and Volker Tresp. 2026b. [Parameter-efficient routed fine-tuning: Mixture-of-experts demands mixture of adaptation modules](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 4439–4457, Rabat, Morocco. Association for Computational Linguistics.
- Zeyu Liu, Tim Dettmers, Xi Lin, Veselin Stoyanov, and Xian Li. 2023. [Towards a unified view of sparse feed-forward network in pretraining large language model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15038–15061.
- Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. 2025. [Autonomy-of-experts models](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41667–41681. PMLR.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.
- Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. [Model cards for model reporting](#). In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* ’19*, page 220–229, New York, NY, USA. Association for Computing Machinery.
- Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2025. [OLMoe: Open mixture-of-experts language models](#). In *The Thirteenth International Conference on Learning Representations*.

- Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. [Adversarial NLI: A new benchmark for natural language understanding](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.
- Nabil Omi, Siddhartha Sen, and Ali Farhadi. 2025. [Load balancing mixture of experts with similarity preserving routers](#). *Preprint*, arXiv:2506.14038.
- Bowen Pan, Yikang Shen, Haokun Liu, Mayank Mishra, Gaoyuan Zhang, Aude Oliva, Colin Raffel, and Rameswar Panda. 2024. [Dense training, sparse inference: Rethinking training of mixture-of-experts language models](#). *Preprint*, arXiv:2404.05567.
- Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. [Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5005–5018.
- Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. [Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale](#). In *International conference on machine learning*, pages 18332–18346. PMLR.
- Inioluwa Deborah Raji, Emily Denton, Emily M. Bender, Alex Hanna, and Amandalynne Paullada. 2021. [AI and the everything in the whole wide world benchmark](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatala, and Yejin Choi. 2021. [Winogrande: An adversarial winograd schema challenge at scale](#). *Communications of the ACM*, 64(9):99–106.
- Girish Sastry, Lennart Heim, Haydn Belfield, Markus Anderljung, Miles Brundage, Julian Hazell, Cullen O’Keefe, Gillian K. Hadfield, Richard Ngo, Konstantin Pilz, George Gor, Emma Bluemke, Sarah Shoker, Janet Egan, Robert F. Trager, Shahar Avin, Adrian Weller, Yoshua Bengio, and Diane Coyle. 2024. [Computing power and the governance of artificial intelligence](#). *Preprint*, arXiv:2402.08797.
- Noam Shazeer. 2020. [Glu variants improve transformer](#). *Preprint*, arXiv:2002.05202.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). In *International Conference on Learning Representations*.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. [Roformer: Enhanced transformer with rotary position embedding](#). *Neurocomputing*, 568:127063.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). *Advances in neural information processing systems*, 30.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [Glue: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.
- Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. 2024a. [Auxiliary-loss-free load balancing strategy for mixture-of-experts](#). *Preprint*, arXiv:2408.15664.
- Ziteng Wang, Jun Zhu, and Jianfei Chen. 2024b. [Re-moe: Fully differentiable mixture-of-experts with relu routing](#). In *The Thirteenth International Conference on Learning Representations*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#). *Transactions on Machine Learning Research*. Survey Certification.
- Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, and 4 others. 2021. [Ethical and social risks of harm from language models](#). *Preprint*, arXiv:2112.04359.
- Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. [Do llamas work in english? on the latent language of multilingual transformers](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394.
- Songhao Wu, Ang Lv, Ruobing Xie, Samm Sun, Di Wang, Rui Yan, and Yankai Lin. 2026. [Union-of-experts: Neurons in mixture-of-experts are secretly](#)

- routers. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36193–36206, San Diego, California, United States. Association for Computational Linguistics.
- Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. [On layer normalization in the transformer architecture](#). In *International conference on machine learning*, pages 10524–10533. PMLR.
- Shen Yan, Xingyan Bin, Sijun Zhang, Yisen Wang, and Zhouchen Lin. 2025. [Tc-moe: Augmenting mixture of experts with ternary expert choice](#). In *The Thirteenth International Conference on Learning Representations*.
- Sikuan Yan, Sicheng Dong, Haotong Wang, Ercong Nie, Yilun Liu, Jinhe Bi, Yingjie Xu, Susanna Schwarzmann, Riccardo Trivisonno, Volker Tresp, and Yunpu Ma. 2026. [Pyramid: Hierarchical multimodal memory for long-horizon video reasoning](#). *Preprint*, arXiv:2605.17065.
- Zhiqing Yang, Yilun Liu, Yunpu Ma, Volker Tresp, and Hinrich Schütze. 2026. [Behavioral and representational evidence of binomial ordering preferences in large language models](#). *Preprint*, arXiv:2606.21645.
- Yanpeng Yu, Haiyue Ma, Krish Agarwal, Nicolai Oswald, Qijing Huang, Hugo Linsenmaier, Chunhui Mei, Ritchie Zhao, Ritika Borkar, Bitu Darvish Rouhani, David Nellans, Ronny Krashinsky, and Anurag Khandelwal. 2025. [Efficient moe serving in the memory-bound regime: Balance activated experts, not tokens](#). *Preprint*, arXiv:2512.09277.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [Hellaswag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.
- Biao Zhang and Rico Sennrich. 2019. [Root mean square layer normalization](#). *Advances in neural information processing systems*, 32.
- Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. [MoEfication: Transformer feed-forward layers are mixtures of experts](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890, Dublin, Ireland. Association for Computational Linguistics.
- Chenqi Zhao, Wenfei Wu, Linhai Song, Yuchen Xu, and Yitao Yuan. 2026. [Fine-grained moe load balancing with linear programming](#). *Preprint*, arXiv:2511.16947.
- Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022. [Mixture-of-experts with expert choice routing](#). *Advances in Neural Information Processing Systems*, 35:7103–7114.
- Barret Zoph. 2022. [Designing effective sparse expert models](#). In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1044–1044. IEEE.

A Load-Balancing Loss

Fedus et al. (2022) first introduced a differentiable auxiliary load-balancing loss to MoE, defined as the scaled dot-product

$$\mathcal{L}_{LB} = \alpha \cdot N \cdot \sum_{i=1}^N f_i \cdot P_i, \quad (18)$$

where f_i is the fraction of tokens dispatched to expert i and P_i is the fraction of router probability allocated to expert i . Since f_i is non-differentiable, the gradient flows through P_i alone, while f_i serves as a fixed per-step weight. The loss is minimized under uniform routing, as $\sum_i f_i P_i = 1/N$ when both vectors are uniform. Our $\mathcal{L}_{EB}$ and $\mathcal{L}_{TB}$ follow the same product-of-averages structure and gradient strategy, extending it to simultaneously encourage uniformity along both the expert and token axes without any centralized routing mechanism. We show that $\mathcal{L}_{EB}$ and $\mathcal{L}_{TB}$ are minimized towards the desired uniform activation state at ρ_∞ . Let

$$f_i = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} f_i(\mathbf{x}), \quad \tilde{g}_i = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} G_i(\mathbf{x}) \quad (19)$$

denote the per-expert mean binary activation and its differentiable proxy. Under the constraint that the adaptive coefficient λ_t pins the mean density $\frac{1}{|\mathcal{E}|} \sum_i f_i = \rho_\infty$, the sum $\sum_i f_i \tilde{g}_i$ is minimized when all terms are equal by the rearrangement inequality, achieved precisely at $f_i = \tilde{g}_i = \rho_\infty$ for all i . While the losses are not written as explicit variance terms, any imbalance where over-activated experts co-occur with larger proxy values increases $\sum_i f_i \tilde{g}_i$ above the baseline ρ_∞^2 attained under perfect balance, since f_i and $\tilde{g}_i$ share a common dependence on $G_i(\mathbf{x})$. The excess above ρ_∞^2 is therefore a monotone function of the covariance between binary activations and their proxies, providing implicit variance penalization through a fully differentiable surrogate without requiring additional normalization terms. An identical argument applies to $\mathcal{L}_{TB}$ by symmetry over the token axis.

B Routing-Free MoE at Deployment

B.1 Expert Parallelism

Deploying MoE at scale often benefits from partitioning experts across multiple devices under *expert parallelism* (EP) (Lepikhin et al., 2021). In

this setting, we trace the full critical path of both architectures and analyze per-regime efficiency.

Consider a single MoE layer distributed to M devices, each hosting N/M experts, and processing a batch of T tokens with hidden dimension D and b bytes per element. We adopt the standard α - β model (Hockney, 1994) that transferring n bytes over one hop incurs latency $\alpha + n/B$, where α is the startup per-hop latency and B is the per-link bandwidth. In standard MoE each token is routed to K experts, and Routing-Free MoE activates an expected $K_{\text{eff}} = \rho_{\infty} \cdot N$ experts per token.

For a standard MoE layer, the router matmul, Softmax, and TopK are strictly sequential and centralized, with costs

$$t_{\text{routing}} = t_{\text{router}} + t_{\text{Softmax}} + t_{\text{TopK}} \\ \propto T \cdot (D + 2) \cdot N. \quad (20)$$

Softmax requires all N logits before any normalization can proceed; TopK requires a full selection pass over N values. Neither can be parallelized across devices, and dispatch cannot begin until both complete. Afterwards it produces a token-to-expert index tensor as dispatch assignment used to pack token buffers, which are sent to multiple target devices via blocking All-to-All:

$$t_{\text{A2A}} = (M-1)\alpha + \frac{K \cdot T \cdot D \cdot b}{M \cdot B}. \quad (21)$$

Expert FFNs run on the received $K \cdot T/N$ tokens per expert with no inter-device communication, with the parallelized per-device cost

$$t_{\text{expert}} \propto 3K \cdot \frac{T}{M} \cdot D \cdot D_{\text{act}}. \quad (22)$$

Their outputs are then scattered back to originating devices with the same buffer geometry and All-to-All t_{A2A} , and the receiving device uses its locally-stored assignment mask to scatter outputs into the correct positions, which is a cheap local memory operation. Total per-layer cost is

$$T_{\text{MoE}} = t_{\text{routing}} + t_{\text{expert}} \\ + 2(M-1)\alpha + 2 \frac{K \cdot T \cdot D \cdot b}{M \cdot B}. \quad (23)$$

Routing-Free MoE starts from a non-blocking All-Gather to broadcast the full token batch:

$$t_{\text{AG}} = (M-1)\alpha + \frac{(M-1) \cdot T \cdot D \cdot b}{M \cdot B}. \quad (24)$$

Each device can immediately begin scoring its N/M local experts on incoming chunks, pipelining scoring with communication. The total scoring cost per device, including applying $\mathbf{x}\mathbf{A}_{\text{gate}}$, norm,

bias, activation and thresholding, is

$$t_{\text{scoring}} \propto T \cdot D \cdot r \cdot \frac{N}{M}, \quad (25)$$

For activated token-expert pairs, $\mathbf{x}\mathbf{A}_{\text{gate},i}$ computed during scoring is directly reused in the FFN forward pass, incurring zero marginal cost. For non-activated tokens computation terminates after the rank- r projection, costing only $T \cdot D \cdot r$ rather than a full FFN pass. The remaining per-device FFN cost for activated pairs is

$$t_{\text{expert}}^* \propto K_{\text{eff}} \cdot \frac{T}{M} \cdot (r + 2D) \cdot D_{\text{act}}. \quad (26)$$

Activated outputs are immediately returned as asynchronous point-to-point messages upon completion, with no barrier required:

$$t_{\text{combine}} = \alpha + \frac{K_{\text{eff}} \cdot T \cdot D \cdot b}{M \cdot B}. \quad (27)$$

Receiving devices accumulate partial sums as results arrive. Total per-layer cost is

$$T_{\text{RFMoE}} = t_{\text{scoring}} + t_{\text{expert}}^* + M\alpha \\ + \frac{(M-1 + K_{\text{eff}}) \cdot T \cdot D \cdot b}{M \cdot B}. \quad (28)$$

Setting $K = K_{\text{eff}}$ for an iso-compute comparison, the ratio of computation cost for routing and expert processing becomes

$$\frac{t_{\text{scoring}} + t_{\text{expert}}^*}{t_{\text{routing}} + t_{\text{expert}}} = \frac{rD + \frac{K}{N}(r + 2D)D_{\text{act}}}{(D + 2)M + \frac{K}{N}(3D)D_{\text{act}}}. \quad (29)$$

Since $r \ll D$ and $M \ll D_{\text{act}}$, this ratio is strictly less than 1, and gets smaller when M grows. Therefore, under expert parallelism, Routing-Free MoE always requires less computation per layer than standard MoE.

For communication, the barrier synchronization saving $\Delta\alpha = (M-2)\alpha$ is strictly positive for all $M \geq 3$, independent of input size or activation ratio. The bandwidth terms differ as

$$\Delta_B = \frac{(K + 1 - M) \cdot T \cdot D \cdot b}{M \cdot B}, \quad (30)$$

which favors Routing-Free MoE when $M < K + 1$. In the prefill stage where T is large, the Δ_B term dominates the communication overhead, and Routing-Free MoE is advantageous when the number of devices M is no greater than the number of activated experts K for each layer, which typically holds in practice for advanced large-scale MoE models (Guo et al., 2025a) under expert-parallel setting at inference. In the decode stage where $T=1$, the bandwidth term becomes negligible and the critical path is instead dominated by α and sequential computational bottlenecks, and Routing-

Setting	T	M	K	Throughput $\uparrow$
<i>Prefill Stage</i>				
MoE	1,024	1	3	22,346.66
MoE	1,024	2	3	18,034.84
MoE	1,024	3	3	18,558.08
MoE	1,024	4	3	18,355.43
RFMoE	1,024	1	3	22,277.77
RFMoE	1,024	2	3	22,268.24
RFMoE	1,024	3	3	21,822.29
RFMoE	1,024	4	3	21,784.77
<i>Decode Stage</i>				
MoE	1	1	3	57.90
MoE	1	2	3	45.65
MoE	1	3	3	45.61
MoE	1	4	3	44.67
RFMoE	1	1	3	52.14
RFMoE	1	2	3	52.05
RFMoE	1	3	3	50.11
RFMoE	1	4	3	49.84

Table 5: Prefill and autoregressive decode tokens-per-second throughput per device under expert parallelism. Prefill process $T=1,024$ tokens in a single forward pass; decode generates 128 tokens autoregressively with $T=1$ per step. $K=3$ denotes TopK for standard MoE and $K_{\text{eff}}=3$ for Routing-Free MoE. All runs use batch size 1 with input tokens sampled randomly from OpenWebText, bfloat16 precision, and report the mean over 10 repetitions after 3 warmup ones.

Free MoE becomes exceptionally well-suited for such high-throughput, low-latency streaming inference scenarios.

To corroborate the theoretical analysis, we evaluate Routing-Free MoE against standard MoE under expert-parallel deployment across varying sequence lengths T , device counts M , and expert activations K for our trained model at scale S in production-representative settings. We isolate the prefill and the decode stages. For each configuration, we measured end-to-end processing time, breaking down contributions from computation, communication, and synchronization overhead.

Table 5 reports throughput under expert parallelism for models at scale S. Standard MoE throughput drops sharply when moving from $M=1$ to $M=2$, with Prefill falls by $\sim 19\%$, and decode falls by $\sim 21\%$, with little recovery at $M=3$ or $M=4$. This reflects the blocking All-to-All dispatches and the sequential, non-parallelizable routing pipeline that must complete before any expert computation begins. In contrast, Routing-Free MoE retains $\geq 97.8\%$ of its single-device prefill throughput and $\geq 95.6\%$ of its decode throughput at $M=4$, consistent with the asynchronous scoring and point-to-point communication pattern de-

scribed above. Although Routing-Free MoE is slightly slower than standard MoE on a single device due to the rank- r gating projection applied to all N experts rather than a single $D \times N$ router matmul, the crossover occurs already at $M=2$, where Routing-Free MoE surpasses MoE by 23% in prefill and 14% in decode throughput. The advantage grows with M . At $M=4$, Routing-Free MoE delivers $1.19\times$ higher prefill throughput and $1.12\times$ higher decode throughput. This confirms the theoretical prediction that the communication and synchronization savings of Routing-Free MoE compound with device count, while its computational cost ratio remains strictly below unity and decreases monotonically in M . The decode stage exhibits particularly graceful scaling because the per-step bandwidth term $T \cdot D \cdot b / (M \cdot B)$ is negligible at $T=1$, leaving the critical path dominated by startup latency α and sequential compute. Here, Routing-Free MoE’s elimination of the centralized routing barrier translates almost entirely into wall-clock savings, highlighting its performance for latency-sensitive autoregressive serving.

B.2 Threshold Adaptation

The design of global post-activation threshold θ provides a lightweight, inference-time mechanism for trading computation against model quality, beyond its role in controlling overall sparsity during training. The routing-free training dynamics naturally encourage each expert to commit decisively to its activation state, which pushes its internal score either well above or well below θ rather than hovering near the boundary. As a result, moderate perturbations to θ at inference time displace only low-confidence, marginally contributing activations, conferring inherent robustness to threshold miscalibration.

Table 6 reports the downstream benchmark average across a sweep of θ at scale S. The most striking observation is the stability of performance despite changes in activation density. Reducing ρ_{eff} from 100% to 4.3%, a more than $20\times$ reduction in expert activations and a 31% drop in FLOPs, degrades the average score by less than two absolute points. This robustness is a direct consequence of the decisive activation patterns learned during routing-free training. Because most expert scores lie far from the decision boundary, sweeping θ over a wide range affects only a small number of uncertain, low-impact activations. Notably, even at extremely low thresholds where nearly all experts

θ	ρ_{eff} (%)	FLOPs	Avg. $\uparrow$
0.1	100.0	120.46M	39.41
0.5	94.8	118.41M	39.12
0.8	61.4	105.40M	39.55
0.9	48.9	100.32M	39.73
1.0	37.8	96.21M	39.80
1.1	29.3	92.97M	39.98
1.2	22.9	90.37M	39.77
1.5	11.1	85.80M	40.77 [†]
1.9	4.3	83.14M	40.23 [†]

Table 6: Effect of the global threshold θ on downstream benchmark average (Avg.) and estimated FLOPs at scale S. ρ_{eff} denotes the empirical mean activation density at θ . [†]Averages at $\theta \geq 1.5$ are driven by the anomalous SST-2 and OBQA spikes (see Figure 8).

are activated, as seen at $\rho_{\text{eff}} \approx 100\%$ with $\theta=0.1$, the model does not suffer from gradient explosion or output instability. Since the activation scores $G_i(\mathbf{x})$ directly scale each expert’s output before aggregation, experts that pass the threshold with low scores contribute proportionally little to the final representation. Lowering θ therefore admits only marginal, low-weight activations rather than introducing equally weighted noise from all experts.

The per-benchmark breakdown in Figure 8 reveals that this aggregate stability is not an artifact of compensating trends. The majority of benchmarks, including HellaSwag, QQP, and QNLI are effectively invariant to θ , confirming that the core representations remain intact even under aggressive sparsification. A small number of tasks exhibit larger sensitivity. SST-2 in particular spikes at $\theta=1.5$, which accounts for most of the elevated aggregate averages at high thresholds. We attribute this to the interaction between task-specific input distributions and the activation patterns of individual experts, rather than a systematic benefit of sparser computation. These results highlight a practical advantage of the routing-free design. The threshold θ serves as a single, transparent knob that allows practitioners to balance accuracy and efficiency at deployment time without retraining, with assurance that moderate adjustments will not disrupt the expert specialization patterns learned during training.

C Statistical Significance Analysis

To support the claim that Routing-Free MoE consistently outperforms the standard MoE baseline, we conduct a formal statistical analysis over the benchmark results reported in Table 1. Each model variant is evaluated on nine benchmarks across three scales, yielding $9 \times 3 = 27$ paired observations, where each pair consists of the accuracy score of

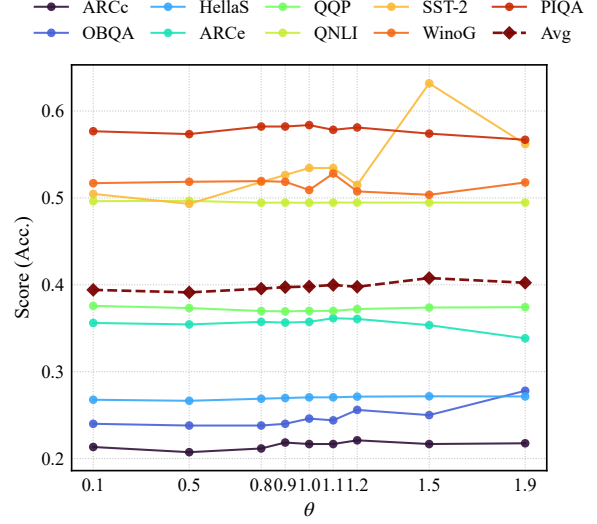


Figure 8: Per-benchmark accuracy across θ at scale S. HellaSwag, QQP, QNLI are nearly invariant to the threshold, while PIQA, ARC-easy, ARC-challenge and Winogrande achieve their best performance around $\rho_{\text{eff}} \approx \rho_{\infty}$. SST-2 and OBQA spike sharply at larger θ , driving the elevated averages in Table 6.

Scale	W/L	Δ Avg.	t -stat	p_t	Cohen’s d
S	5 / 4	+0.80	1.481	0.089	0.494
M	6 / 3	+0.76	0.764	0.233	0.255
L	5 / 4	+0.76	1.184	0.135	0.395
Overall	16 / 11	+0.77	1.858	0.037	0.358

Table 7: Paired statistical tests comparing Routing-Free MoE against the standard MoE baseline across 9 benchmarks. p -values are one-sided.

Routing-Free MoE and MoE on the same benchmark at the same scale.

We treat each (scale, benchmark) pair as a matched observation. For each pair we compute the signed improvement $\Delta_i = \text{RFMoE}_i - \text{MoE}_i$ in percentage points (pp). We test the one-sided null hypothesis $H_0: \mu_{\Delta} \leq 0$ against $H_1: \mu_{\Delta} > 0$. We apply a one-sided paired t -test, which evaluates whether the mean improvement is significantly positive under an approximate normality assumption. Effect size is quantified via Cohen’s d computed on the paired differences, and win rate is reported as a descriptive statistic.

Table 7 reports statistics for each scale. Taking together all observations ($n = 27$), Routing-Free MoE achieves a positive improvement of +0.77 pp, and the one-sided paired t -test yields $t = 1.858$, $p = 0.037$, and Cohen’s $d = 0.36$, rejecting H_0 at the $\alpha = 0.05$ level. Beyond downstream task accuracy, Routing-Free MoE also achieves consistently lower perplexity than the MoE baseline at

every scale, with -12.2% at S, -11.7% at M, and -18.8% at L, suggesting that the quality of language modeling improves consistently with scale under Routing-Free MoE’s design, independent of the downstream accuracy results.

Taken together, the statistical analysis supports the conclusion that Routing-Free MoE produces a reliable improvement over the standard MoE baseline. The effect is consistent in direction across all three scales and nine benchmarks, reaches statistical significance when observations are pooled with paired t -test $p = 0.037$, along with substantial perplexity reduction at every scale.

D Additional Discussion

D.1 Load-Balancing

A recent line of work explored auxiliary-loss-free load balancing (DeepSeek-AI et al., 2025; Guo et al., 2025a). Our baselines and Routing-Free MoE all employ auxiliary losses following standard practice, ensuring a controlled comparison across routing architectures. Since auxiliary-loss-free balancing is orthogonal to the routing mechanism itself, any such technique could in principle be applied to standard MoE, AoE, ReMoE, and Routing-Free MoE alike. Introducing it for one method but not others would confound the comparison, while applying it uniformly across all baselines would constitute a substantial engineering effort that lies beyond the scope of this work. We therefore focus on the architectural distinction between routing-based and routing-free expert selection, which is the central contribution of this paper, and leave the integration of auxiliary-loss-free balancing as beyond the scope of this work.

Similarly, since all architectures compared in this work operate under the Token Choice load balancing paradigm, which is the dominant setting in the MoE literature (Muennighoff et al., 2025), we choose not to include Expert Choice routing as a direct baseline. Instead, our configurable μ -interpolation between token-balancing and expert-balancing losses allows Routing-Free MoE to smoothly interpolate the spectrum between token-balancing and expert-balancing behavior within a unified framework. This design choice enables a thorough internal comparison across balancing strategies, as demonstrated in our ablation studies, without introducing the confound of an entirely different assignment paradigm.

D.2 Per-Layer and Global Density

In Figure 6, without any explicit supervision, the model develops a striking three-stage structure, with early layers showing high but rapidly decreasing activation with large variance, middle layers exhibiting a slow monotonic climb with low variance, and late layers displaying a sharp rise in both activation level and variance. This emergent pattern coincides closely with interpretability findings on the heterogeneous functional roles that layers naturally develop in LLMs (Geva et al., 2021; Gao et al., 2024a; Wendler et al., 2024; Jiao et al., 2024), with early layers converting input from token space to concept space, middle layers processing them, and late layers preparing input-dependent outputs back to token space. We argue that the performance gain from global density constraints stems precisely from removing the per-layer inductive bias. Enforcing identical sparsity at every depth suppresses the compute-hungry layers that naturally benefit from activating more experts, while simultaneously forcing unnecessary activations in layers where sparse representations suffice. Once this bias is lifted, the model is free to self-organize into a more effective, functionally aligned activation structure.

E Additional Experiment Results

Arch.	Config.	r	L	D	D_{act}	Size	FLOPs	λ	η	μ	α	Loss	Val. Loss	PPL
MoE	Top3/12	n/a	12	512	128	92.44M	90.93M	n/a	n/a	n/a	$5e^{-4}$	3.804	3.667	39.13
MoE	Top3/12	n/a	12	512	128	92.44M	90.93M	n/a	n/a	n/a	$1e^{-3}$	<u>3.595</u>	<u>3.441</u>	<u>31.22</u>
MoE	Top3/12	n/a	12	512	128	92.44M	90.93M	n/a	n/a	n/a	$2e^{-3}$	~~5.584~~	~~5.366~~	~~214.0~~
AoE	Top3/12	16	12	512	128	93.85M	88.57M	n/a	n/a	n/a	$1e^{-3}$	3.440	3.401	30.00
AoE	Top3/12	32	12	512	128	95.32M	91.08M	n/a	n/a	n/a	$1e^{-3}$	3.450	3.411	30.31
ReMoE	Top3/12	n/a	12	512	128	92.44M	90.93M	$1e^{-8}$	0.2	n/a	$1e^{-3}$	3.521	3.389	29.60
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.005	0.5	$5e^{-4}$	3.579	3.620	37.34
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.05	0.5	$5e^{-4}$	3.552	3.600	36.60
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.08	0.5	$5e^{-4}$	3.537	3.644	38.24
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.1	0.5	$5e^{-4}$	3.571	3.612	37.04
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	3.309	3.358	28.73
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.02	0.5	$2e^{-3}$	<u>3.261</u>	<u>3.311</u>	<u>27.42</u>
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.02	0.5	$5e^{-3}$	~~7.097~~	~~6.971~~	~~4065~~
RFMoE	$\rho_{\infty}=1/4$	8	12	512	128	93.11M	87.32M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	3.332	3.373	29.16
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	3.309	3.358	28.74
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	3.290	3.344	28.34
RFMoE	$\rho_{\infty}=1/4$	64	12	512	128	98.27M	96.09M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	3.290	3.341	28.24
RFMoE	$\rho_{\infty}=1/4$	8	12	512	128	93.11M	87.32M	$1e^{-10}$	0.02	0.5	$2e^{-3}$	3.717	3.699	40.41
RFMoE	$\rho_{\infty}=1/4$	16	12	512	128	93.85M	88.57M	$1e^{-10}$	0.02	0.5	$2e^{-3}$	3.311	3.261	26.08
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	0.5	$2e^{-3}$	~~6.435~~	~~6.564~~	~~709.1~~
RFMoE	$\rho_{\infty}=1/4$	64	12	512	128	98.27M	96.09M	$1e^{-10}$	0.02	0.5	$2e^{-3}$	3.548	3.516	33.65
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	0.0	$1e^{-3}$	3.481	3.347	28.41
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	0.2	$1e^{-3}$	3.483	3.345	28.35
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	0.8	$1e^{-3}$	3.488	3.346	28.38
RFMoE	$\rho_{\infty}=1/4$	32	12	512	128	95.32M	91.08M	$1e^{-10}$	0.02	1.0	$1e^{-3}$	3.485	3.347	28.43
MoE	Top4/16	n/a	24	768	192	289.92M	247.95M	n/a	n/a	n/a	$5e^{-4}$	3.808	3.300	27.11
MoE	Top4/16	n/a	24	768	192	289.92M	247.95M	n/a	n/a	n/a	$1e^{-3}$	<u>3.726</u>	<u>3.219</u>	<u>25.00</u>
RFMoE	$\rho_{\infty}=1/4$	48	24	768	192	307.30M	249.17M	$1e^{-10}$	0.02	0.5	$5e^{-4}$	3.360	3.292	26.90
RFMoE	$\rho_{\infty}=1/4$	48	24	768	192	307.30M	249.17M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	<u>3.166</u>	<u>3.095</u>	<u>22.08</u>
MoE	Top6/24	n/a	32	1024	256	808.42M	608.38M	n/a	n/a	n/a	$5e^{-4}$	<u>3.947</u>	<u>3.202</u>	<u>24.58</u>
MoE	Top6/24	n/a	32	1024	256	808.42M	608.38M	n/a	n/a	n/a	$8e^{-4}$	~~4.493~~	~~3.768~~	~~43.29~~
MoE	Top6/24	n/a	32	1024	256	808.42M	608.38M	n/a	n/a	n/a	$1e^{-3}$	~~8.532~~	~~8.104~~	~~3309~~
RFMoE	$\rho_{\infty}=1/4$	64	32	1024	256	870.60M	613.19M	$1e^{-10}$	0.02	0.5	$5e^{-4}$	3.179	3.107	22.36
RFMoE	$\rho_{\infty}=1/4$	64	32	1024	256	870.60M	613.19M	$1e^{-10}$	0.02	0.5	$8e^{-4}$	<u>3.076</u>	<u>2.994</u>	<u>19.97</u>
RFMoE	$\rho_{\infty}=1/4$	64	32	1024	256	870.60M	613.19M	$1e^{-10}$	0.02	0.5	$1e^{-3}$	~~5.448~~	~~3.548~~	~~34.73~~

Table 8: **Performance of standard MoE, AoE, ReMoE, and Routing-Free MoE across configurations.** Each model was trained on OpenWebText (Gokaslan et al., 2019). Results underlined correspond to those reported in Table 1. Entries marked as n/a indicate fields not applicable to the respective architectural design. Strikethrough values denote experiments that failed to complete their training due to gradient explosion or inferior performance.